

**Carnegie Mellon  
Software Engineering Institute**

---

# **Product Line Production Planning for the Home Integration System Example**

Gary Chastek  
Patrick Donohoe  
John D. McGregor

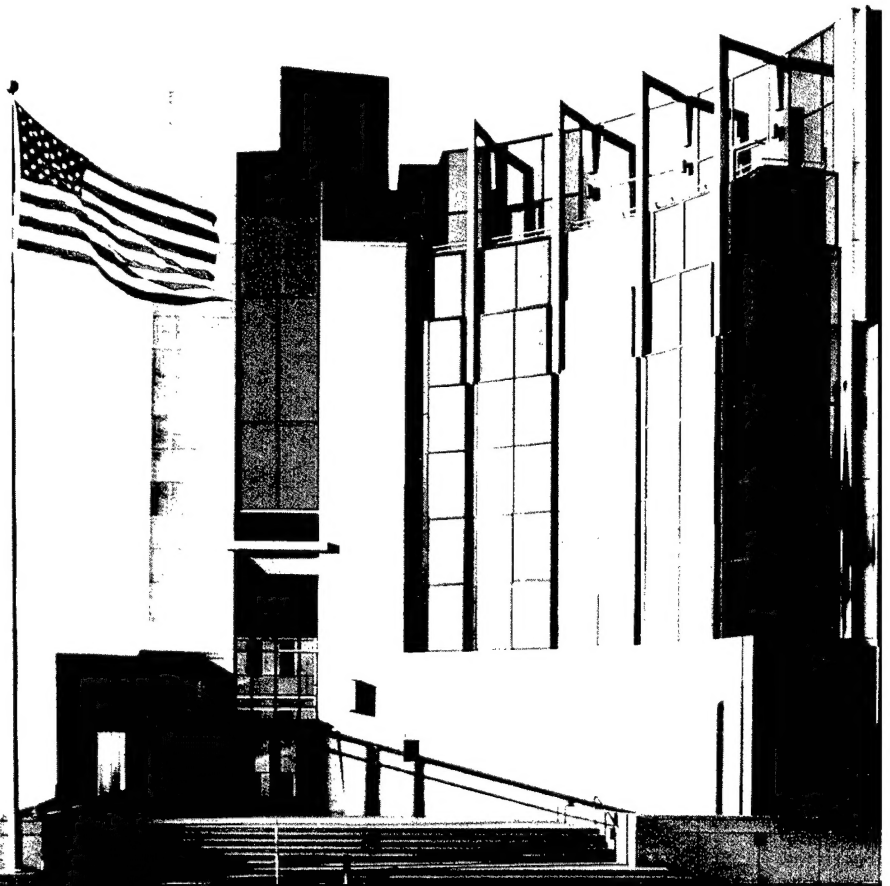
*September 2002*

**Product Line Practice Initiative**

20021231 104

Unlimited distribution subject to the copyright.

**Technical Note**  
CMU/SEI-2002-TN-029



# **Product Line Production Planning for the Home Integration System Example**

Gary Chastek  
Patrick Donohoe  
John D. McGregor

*September 2002*

**Product Line Practice Initiative**

Unlimited distribution subject to the copyright.

**Technical Note**  
CMU/SEI-2002-TN-029

*AQMO3-03-0628*

The Software Engineering Institute is a federally funded research and development center sponsored by the U.S. Department of Defense.

Copyright 2002 by Carnegie Mellon University.

#### NO WARRANTY

THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

Use of any trademarks in this report is not intended in any way to infringe on the rights of the trademark holder.

Internal use. Permission to reproduce this document and to prepare derivative works from this document for internal use is granted, provided the copyright and "No Warranty" statements are included with all reproductions and derivative works.

External use. Requests for permission to reproduce this document or prepare derivative works of this document for external and commercial use should be addressed to the SEI Licensing Agent.

This work was created in the performance of Federal Government Contract Number F19628-00-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 252.227-7013.

For information about purchasing paper copies of SEI reports, please visit the publications portion of our Web site (<http://www.sei.cmu.edu/publications/pubweb.html>).

---

## Contents

<b>Abstract.....</b>	<b>vii</b>
<b>1 Introduction .....</b>	<b>1</b>
1.1 Production Strategies and Plans for Product Lines .....	1
1.2 Home Integration Systems .....	2
<b>2 Connect'Em: A Networking Company.....</b>	<b>4</b>
<b>3 Production Planning for the Connect'Em Company .....</b>	<b>6</b>
3.1 Introduction .....	6
3.1.1 Production Context .....	6
3.1.2 Audience.....	7
3.1.3 Qualifications .....	7
3.2 Strategic View of Product Development.....	7
3.2.1 Assumptions .....	7
3.2.2 Qualities.....	7
3.2.3 Products Possible from Available Assets .....	8
3.2.4 Production Strategy .....	9
3.3 Overview of Available Core Assets .....	10
3.3.1 Basic Inputs and Dependencies.....	10
3.3.2 Variations.....	11
3.4 Detailed Production Process .....	12
3.4.1 Requirements Engineering.....	12
3.4.2 Architecture Definition .....	12
3.4.3 Architecture Evaluation .....	13
3.4.4 Component Development .....	13
3.4.5 Testing.....	13
3.4.6 Software System Integration .....	13
3.5 Configuration Management .....	14
3.6 Tailoring the Production Plan .....	14
3.6.1 Product Production .....	14
3.7 Management Information.....	14
3.7.1 Bill of Materials .....	14
3.7.2 Production Resources.....	15
3.7.3 Schedule.....	16

3.7.4	Product-Specific Details .....	16
3.7.5	Metrics.....	16
<b>4</b>	<b>Protect'Em: A Home Security Company.....</b>	<b>18</b>
4.1	Production Plan Differences.....	18
<b>5</b>	<b>Fleece'Em: A Home Automation Corporation .....</b>	<b>21</b>
5.1	Production Plan Differences.....	21
<b>6</b>	<b>Summary .....</b>	<b>24</b>
<b>Appendix</b>	<b>Outline of a Production Plan.....</b>	<b>26</b>
<b>References</b> .....		<b>29</b>

---

## List of Figures

Figure 1: The OSGi Conceptual Model.....	4
--	---



---

## List of Tables

Table 1:	Partial Listing of Device Packages and Quality Attributes.....	8
Table 2:	Service Packages' Association with HIS Products.....	9
Table 3:	Bill of Materials for a Connect'Em Product .....	15
Table 4:	Comparison of the Three Production Plans .....	25



---

## **Abstract**

A production plan is a description of how a software product line organization builds products in a product line. This technical note examines the significant characteristics of the production plans of three hypothetical organizations that create product lines of home integration systems. Such systems enable homeowners to access and control equipment in their homes such as climate control and security systems. The plan for one of the organizations is presented in some detail, and the plans for the other two are described in terms of their differences from the first plan. The purpose of this note is to show how influences such as an organization's business goals, production strategy, and experience in product lines can lead to very different approaches to building products.



---

# 1 Introduction

This technical note describes the essentials of the production plans developed by three hypothetical software product line organizations that are entering the home integration systems (HISs) market. It is intended for core asset and product developers who are familiar with the companion report *Guidelines for Developing a Product Line Production Plan* [Chastek 02]. This note elaborates on the guidance provided in that report by describing the production plan of one of the hypothetical organizations in some detail and how it differs from the plans of the other two organizations. The production-planning approaches used aren't necessarily the only ones they could have chosen; the intent of the examples is to show how the information to be communicated to product developers varies along several dimensions. Rather than providing detailed production plans for the three organizations, this document focuses on showing how an organization's business goals, production strategy, and experience in product lines affect how products are created.

The remainder of this section provides a brief overview of the concepts discussed in this technical note<sup>1</sup> and an introduction to the HIS market. Sections 2, 3, and 4 describe the example organizations and how they approach production planning for their HIS product lines. Each organization has experience with parts of the HIS domain and is entering the HIS market for the first time. Each description addresses the organizations' current situation, market and business goals, domain expertise, developer expertise, and reasons for entering the HIS market.

Sections 2 and 3 describe the Connect'Em organization and its HIS production plan in detail. Section 4 describes the Protect'Em organization and explains how and why its HIS production plan differs from Connect'Em's. Those differences are explained by how the organization's business goals map to the required qualities of its production system for product lines. Section 5 provides similar information for the Fleece'Em organization. Section 6 summarizes this technical note.

## 1.1 Production Strategies and Plans for Product Lines

Products in a product line are built from the product line's core assets and their attached processes [Clements 02]. These assets include requirements, architecture, components, test cases and plans, documentation, schedules, and budgets. A core asset's attached process describes how the asset is to be used in the building of products. The production plan tells product developers how the assets and attached processes are applied to build a specific

---

<sup>1</sup> Chastek and McGregor provide a more detailed discussion of these concepts [Chastek 02].

product. It coordinates the efforts of managers, product developers, testers, and clients.<sup>2</sup> The plan links the information provided by the product requirements, business case, architecture description, component specifications, asset-use processes, and other sources such as user manuals.

The production plan evolves from the production strategy. That strategy begins as an informal notion in the business case, evolves concurrently with the core assets, and is documented ultimately in the production plan. The production strategy is based on the goals of the product line and specifies the techniques and conditions for product development that support those goals. For example, part of the production strategy may be to purchase several components that would be too expensive for the organization to develop. The production plan would identify those components and include instructions for tailoring them for a given product (for example, by specifying the product-specific parameter values to apply to the generic tailoring instructions that accompany the component).

## 1.2 Home Integration Systems

A home integration system (HIS) enables homeowners to access, control, and integrate equipment in their homes such as those listed below [Bachmann 00, Chastek 01]:

- climate control systems—heating and cooling
- security systems—intruder, fire, and flood detection and response
- entertainment systems—televisions, radios, and music-playing devices
- personal computers
- telecommunications systems—remote access, status display, and control
- major home appliances

Typically, HISs are

- long-lived, lasting for the lifetime of a house
- upgradeable, enabling devices to be added or removed
- modifiable, can be expanded into related markets (e.g., office or apartment buildings)

---

<sup>2</sup> A fully automated process would eliminate these efforts entirely. The assumption here is that most organizations will have only a partly automated production process, and that the production plan will provide the overall guidance that spans both manual and automated processes.

- reliable
  - When devices are added or removed, the HIS remains operational.
  - The failure of a single device does not crash the HIS.
  - If the HIS crashes, the devices it controls must still continue to work.
  - The HIS is at least as reliable as the devices attached to it.
- interoperable and able to control multiple devices produced by multiple suppliers
- secure, offering
  - multiple levels of authentication for local and remote users
  - confidentiality to support multiple users
- usable
  - The average homeowner does not need special skills to operate HISs.
  - HISs are tolerant of human error.

HISs represent a projected multibillion-dollar market and offer an opportunity for an organization with strong experience and expertise in a portion of the HIS arena (e.g., integration, networks, devices, home or office security systems) to expand into a new market. Market opportunities include

- device and network hardware
- software device drivers
- platform software for integrating services
- network software
- intelligent user interfaces

The HIS market is relatively new and immature, and is changing rapidly as new devices and manufacturers continually appear. Most homeowners have no experience with an HIS and may be unsure of its value or their need for it. The consumer market for HISs is also quite varied. One customer might want only the ability to turn on the air conditioner at night. Another customer might want to track people as they move through the house and adjust settings to their individual tastes. Another consideration is that as customers become more sophisticated users, their requirements will change, perhaps dramatically.

## 2 Connect'Em: A Networking Company

Connect'Em is a networking company that has been producing connection solutions for over 10 years. The company has specialized in embedded applications such as its router based on the Common Object Request Broker Architecture (CORBA). It has developed solutions using a number of hardware and software technologies. The company's software expertise is in providing efficient implementations of networking protocols and matching the qualities of each implementation to the technology selected for its corresponding product. Connect'Em's products range over Bluetooth, IEEE 802.11 LAN, Jini, and broadband IEEE 1394 protocols.

Connect'Em, which has worked with both wired and wireless protocols, was asked to provide a connection solution for the security, telephone, and climate control systems for an industrial client. That client requested that the Open Systems Gateway Initiative (OSGi)<sup>3</sup> protocol be used because of the wide range of device types it supports and to allow for ease of configuration in specific installations [OSGi 02]. The OSGi standard provides one of the most comprehensive solutions by integrating communication from users to servers. Figure 1 shows the conceptual model for the OSGi standard and how the modular nature of an OSGi-compliant system allows individual sets of use cases to be associated with each individual service.

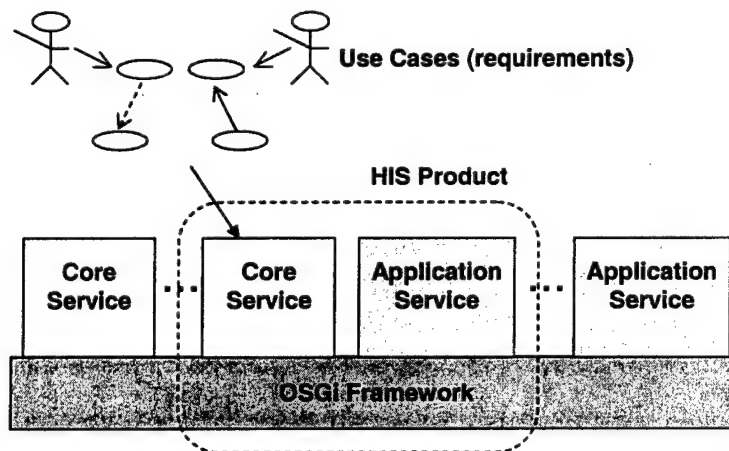


Figure 1: The OSGi Conceptual Model

<sup>3</sup> Information on the OSGi is provided at its Web site at <<http://www.osgi.org>>.

While creating the requested product, Connect'Em's development team investigated the OSGi standard thoroughly. The team identified an emerging market opportunity, HISs, that would allow Connect'Em to leverage the industrial experience that it gained on the current project to address the much larger home market.

Connect'Em has been using a product line approach for hardware for about seven years and, within the last two years, has been using a software product line approach to the software portion of its products. The company currently has three product lines: a series of routers for private telephone systems used in small businesses and two series of routers for wireless networks used in manufacturing process-control applications. The main variation among the products is the protocol that formats and processes data. A second variation is the volume of traffic that each product can carry. The company believes that this experience in product lines can be leveraged to advantage for the new venture.

Connect'Em will address the new opportunity in HISs by creating a new product line and populate the product line by reengineering as many assets from its industrial products and other product lines as possible. The standard OSGi architecture and the modified version of it that Connect'Em developed for its industrial client will be the starting points for the product line architecture.

Connect'Em's products are based on the conceptual model shown in Figure 1. Each product consists of a central nervous system (CNS), devices and their drivers, and a set of services [Bachmann 00]. The CNS will be constructed as an integral part of the OSGi framework in the architecture. Connect'Em will leverage its expertise in networking to provide clients with choices of basic communication protocols, varying the capacity of the system. Basic system packages will include the software necessary to add new devices to an existing control system. They will also include a variety of software services that take advantage of the devices that can be connected to the system.

Connect'Em does not have a marketing department that interacts directly with homeowners since its customers traditionally have been businesses. The company decided to sell to homeowners through large home-improvement chains to reduce the risk of its expansion into the home market. Connect'Em will depend on representatives of these companies to understand their customers' skill levels and to make professional installation available to those who want it. Connect'Em will sell a basic starter system that can be upgraded, in addition to a series of increasingly sophisticated systems and a set of accessories.

---

## **3 Production Planning for the Connect'Em Company**

Production plans are detailed documents that specify the overall production strategy, the materials to be used, and the schedule of activities for producing the product. The following is a description of the contents of Connect'Em's production plan and the rationale for the choices that were made. It is not intended to be an actual production plan.

### **3.1 Introduction**

#### **3.1.1 Production Context**

The products being constructed in the Connect'Em product line are complete HISs. Each of these products consists of the CNS, individual devices and their drivers, and a set of compatible services. These services provide the system user with control over a set of devices (such as appliances) and household systems (such as heating and air conditioning units). Products will be delivered to retailers as a "bundle" that includes a core system and a set of service packages. The core system includes a version of the CNS and documentation that describes possible systems that could be built using the included packages. The products allow for additional devices and services, which use a plug-and-play approach and don't require systems expertise, to be added in the field.

The production plan for a complete HIS describes the steps of

- identifying the set of services that cover the functions required for the product and that are compatible with the required qualities
- selecting a version of the CNS that has the capacity to support the required number of devices
- testing various configurations of the CNS and services to determine that the system achieves the required qualities

The plan also describes how new services and device drivers can be produced. Each service is developed and certified by a product development team. For each specific product, the product development team will modify this general production plan to include details specific to that product. The sections below describe how the general plan should be modified for a specific product.

### **3.1.2 Audience**

The product developers at Connect'Em are the primary audience of a production plan. The plan provides directions on how to build their assigned product from the core assets. The core asset developers and the product line managers are secondary audiences for the plan. The plan places certain requirements on the assets that will support the production strategy. These requirements include the need for a system configuration and prediction tool, and for a specification notation that allows product developers to understand the limitations on the individual devices used in the system. The core asset developers have taken these implicit requirements into account when developing the core assets. The product line managers participate in developing the product-specific production plan from the general production plan.

### **3.1.3 Qualifications**

Users of the Connect'Em production plan are expected to be familiar with the HIS domain and the OSGi standard. In addition, they should be familiar with the product line's concept of operations (CONOPS) and with the operation of the product line organization.

## **3.2 Strategic View of Product Development**

This section of the plan documents the production strategy. HISs are very modular; customers purchase exactly the services they need. Products are configured in the field by installers or adventurous do-it-yourselfers. The production strategy—modular product development—mirrors the product's structure.

### **3.2.1 Assumptions**

Connect'Em's production plan has been created with the following assumptions about the product line in mind:

- The HIS domain is immature and evolving. The components used to build systems will be modified often.
- Most buyers of HISs do not yet understand their full potential. New products will be identified through experimentation with available core assets and added to the product line.
- The core assets have been constructed with the plug-and-play production strategy in mind.

### **3.2.2 Qualities**

The two qualities that are most important to the developer of a product in the Connect'Em product line are modularity and configurability. The plug-and-play nature of the system

requires modularity. When a device is removed from the physical system, the service component associated with the device is automatically removed from the system, thereby reducing its complexity. The OSGi architecture and the “package” concept being used to encapsulate services and devices support the modular nature of the product.

The products in the product line must also be configurable. Different customer priorities and differences in hardware make each deployed system unique. The production plan will guide the product developer in investigating different configurations and determining whether they are suitable.

The third most important quality for the product developer is performance, since real-time sensors are being read. It is possible to load a system with so many devices that readings become inaccurate. However, since the system is not required to react to events in hard-real-time, performance is less important than the ability to configure a system that meets a client’s needs.

From the customer’s point of view, reliability is the most important quality. The system includes the option of a battery backup to increase reliability. The production plan provides more precise information about the operational profile of the product and allows more focused testing to ensure reliability. The plan also prescribes a conformance-testing process to certify the reliability of any vendor-supplied driver.

Connect’Em’s product line architects have achieved a balance among the required qualities. Later sections of the plan describe how to verify that a product possesses these required qualities.

### **3.2.3 Products Possible from Available Assets**

The scope of Connect’Em’s product line defines a range of products but does not provide the level of detail needed by the product developers. The production plan takes the details of the core assets, including their attached processes, and provides a framework that guides the product developers during product assembly.

The available assets are the CNS, a set of service packages, and a set of device packages.<sup>4</sup> As new device and service packages become available, they are rated on a set of qualities that represent the runtime qualities of a composed system. Table 1 shows a partial listing of how the device packages are rated relative to quality attributes.

---

<sup>4</sup> Mini-projects are organized around each device. The project team is chartered to analyze the driver acquired with the hardware device and to design the modifications needed to incorporate the device into a Connect’Em product. The result is a device package that will usually be part of multiple products.

Table 1: Partial Listing of Device Packages and Quality Attributes

Product	SecurSystem Model A	SecurSystem Model B	Appliance Controller 110	Appliance Controller 220
Performance Rating	Low	High	Low	Medium
Capacity Rating	Medium	High	Low	High

This initial product line consists of three products. The Econo-HIS is the cheapest, lowest capacity system. The products increase in capacity, granularity of control, and price from Econo-HIS to Lux-HIS. Table 2 illustrates how several service packages are associated with those HIS products.

Table 2: Service Packages' Association with HIS Products

Product	SecurSystem Model A	SecurSystem Model B	Appliance Controller 110	Appliance Controller 220
Econo-HIS	X		X	
Mid-HIS		X	X	
Lux-HIS	X	X	X	X

### 3.2.4 Production Strategy

The strategy for producing products in this product line is *assemble and configure*. First, the product developers assemble a use case model from the modular sets of use cases and then assemble the corresponding services and drivers into a product. Second, they configure the CNS to provide the appropriate precedence rules among the events produced by the services and to provide default sensing levels.

The *assemble* portion of the strategy is intended to support experimentation with new products and allow upgrades in the field. This motivated the product line architects to design the service assets to plug directly into the CNS and automatically interact with devices that support certain interfaces.

For example, the security service is designed to interact with devices that can be set to detect. In response to user input, the security service issues a "set to detect" event. All devices that implement the security interface respond to this event by activating their sensors. When a sensor is in the "detect" state and the condition it measures changes, the sensor issues a "detected" event. The CNS receives this event and applies currently active rules that determine its response. The CNS may deliver this event to the security service, which takes the appropriate action. Alternatively, it may elect to delete the event if the system is currently in "sensor test" mode.

The *configure* portion of the strategy complements the assemble portion by ensuring that the CNS knows how to deal with such events. The product developer's main role is to ensure that the rule set in the CNS is complete and correct. Because of the life-critical nature of some of these systems, the strategy calls for an intensive test suite that presents various scenarios to determine that life-threatening interactions do not occur. These tests are run during production.

### **3.3 Overview of Available Core Assets**

Product developers have a complete asset base available to them. The associated documentation provides an overview of the core assets and directs the developer to the attached processes for further information. The developers should have read the product line scope document to gain a high-level understanding of the products. They should also have read the CONOPS to understand the relationships among the groups in the organization and the procedures for communicating with the core asset developers.

The following subsections provide more details on some of the available core assets.

#### **3.3.1 Basic Inputs and Dependencies**

##### **3.3.1.1 Architecture**

Bachmann describes the architecture of a basic HIS [Bachman 00]. The Connect'Em architecture incorporates the OSGi standard architecture as the key abstraction of the fundamental system. The OSGi standard defines an open, extensible architecture that allows the introduction of additional services and is a more comprehensive standard than others (e.g., the HomePlug standard [HomePlug 02] which is limited to powerline devices). OSGi services can interact over both wired and wireless devices.

##### **3.3.1.2 Code Assets**

The code assets consist of implementations of

- the CNS
- pluggable services
- devices and their associated software drivers
- vendor-supplied drivers
- a variety of test harnesses

The code assets have been implemented with the Prediction-Enabled Component Technology (PECT) [Hissam 01]. This technology allows certain qualities of a finished system to be predicted from the measurements of the components that will comprise it. Those drivers

supplied directly by vendors do not conform to this technique and will not be included in quality-level predictions.

#### **3.3.1.3 Non-Code Assets**

The non-code assets include

- a use case model and a set of quality scenarios
- a mapping document that illustrates how Connect'Em's architecture implements the OSGi standard
- test plans linked to requirement sets and quality scenarios
- test results from previous tests
- modular documentation

The non-code assets are intended either to support product development or to be included in the deliverable for the customer.

#### **3.3.1.4 Tools**

The core asset team has constructed a number of tools for the product developers. The most important one is the prediction and modeling tool that is also intended for use in the field. That tool allows the product developers to check configurations of devices, services, and rule sets for possible conflicts.

The core asset team has explicitly not provided an automated product assembly tool. There are too few products in the product line, and the range of devices that may be included in each product is too broad to make that type of automation a useful idea. The plug-and-play technology should make product assembly easy for the product developers.

### **3.3.2 Variations**

The available core assets determine the range of variations that can be supported. The production plan gives an overview of the types of variations that are available and leaves the details to the attached processes of the core assets.

*Granularity of control:* The more expensive systems offer the customer greater control over the events in the home. For example, when fire is detected by the Econo-HIS product, it is impossible to determine exactly where the fire is in the house. With the Lux-HIS product, fire zones are established, and different responses to a fire can be programmed for each zone.

*Response time:* The more expensive systems feature a more powerful controller that handles events more rapidly. The upgraded controllers and sensors are also more reliable.

*User interfaces:* The Connect'Em product line has a variety of methods by which users interact with systems. Standard input/output interfaces include a central control panel and remote controls. The more expensive systems offer personal computer (PC), Web, and personal digital assistant (PDA) interfaces. Pagers and email clients, for example, may receive notifications and emergency messages but prohibit user input.

*Runtime environment:* Three main options are available for the runtime environment:

1. Connect'Em's original runtime environment, which is a real-time operating system with error handlers. This option is the easiest one for HIS owners to maintain.
2. a Linux-based runtime environment that includes a secure Web server. This option arose from new user interface options (such as the Web interface) that pointed out the need for greater security.
3. a lighter-weight environment based on a popular PDA operating system that participates in a unidirectional email protocol but not the interactive Web protocol. This option also arose from new user interface options.

### **3.4 Detailed Production Process**

As Chastek and McGregor describe [Chastek 02], the production process is structured according to the Product Builder pattern [Clements 02]. The pattern elements form a major portion of the production process and are described below. These steps must accomplish the three tasks outlined in Section 3.1.1. In an actual production plan, some of these steps would be eliminated depending on the specific requirements.

#### **3.4.1 Requirements Engineering**

Product engineering begins with selecting the services (features) that should be included for the specific product and using this information to determine if the proposed product is within the product line's scope. This activity is performed by the product-planning group based on input from the marketing and technology-forecasting groups. From the feature list, a detailed set of requirements is constructed in the form of use cases [Jacobson 99]. Figure 1 shows that a set of use cases is associated with each service. These are included in the service package description. The requirements engineer can trace from the selected use cases directly to the services that will meet the customers' needs and from there to the components that implement the services.

#### **3.4.2 Architecture Definition**

Little if any architecture definition work is needed for a specific product. The CNS hides much of the OSGi standard architecture from the product developer. The basic architecture is designed to be extensible and to have devices added over time. Only devices that cannot be

managed by an OSGi-conformant driver require special architecture definition work. These types of devices are outside the scope defined for the Connect'Em product line.

### **3.4.3 Architecture Evaluation**

The architecture for a product is evaluated only if the architecture has been modified. The focus of the evaluation is to ensure that the selected services, attached devices, and revised interfaces are compatible.

### **3.4.4 Component Development**

Little if any component development is needed for a specific product. New components are developed when a component is acquired that does not implement the driver protocol required for plug-and-play capability. The new component encapsulates the acquired component and the "glue" code needed to integrate the new asset with the existing set. The core asset team uses the PECT-based tool (see Section 3.3.1.2) to configure sample systems to check for faulty interactions between the new component and existing components. The tool allows product developers to visually select the devices and services, and to place them in the sample system, connected in the way they would be in the real system. In addition, the tool provides feedback to the tool user about potential performance bottlenecks or long wiring runs that impede performance. Core asset developers also use the tool as they test various configurations of new assets. The product developers use the tool to ensure that any modified components provide acceptable quality values.

### **3.4.5 Testing**

All the code assets will have been thoroughly tested by the core asset builders during component development, and the non-code assets will have been inspected. Product developers must achieve the same levels of test coverage as the core asset developers for any new component development. Even if no new development is performed, the specific product must be tested as an entity. Test coverage is measured by the number of different combinations of service packages that are evaluated in the working system. The Orthogonal Array Testing System is used to reduce the number of test configurations that must be executed to ensure adequate coverage [McGregor 01].

### **3.4.6 Software System Integration**

The product builders integrate the selected services and configure the product. The CNS senses the services as they are added to the product; however, in some cases, additional configuration is required, particularly if two services have identical priorities. The modular documentation pieces are integrated into a single document. The product builders check the core asset test reports to determine whether the specific set of services has been tested

previously. If not, the team uses the modular service tests for core assets to produce and execute test scenarios for the specific product.

### **3.5 Configuration Management**

The production plan describes the specific configuration management (CM) file or branching structure to use with the product being built. If the product team adds or modifies software that is specific to its product, the team is responsible for creating the appropriate groupings in the CM system. That system provides the traceability between the requirements, components, and subsystems of the HIS. The conceptual model in Figure 1 shows an association between a set of use cases that describe requirements, components that implement specific core and application services, and an HIS that is an aggregation of the CNS and a set of components.

### **3.6 Tailoring the Production Plan**

Each product development team customizes the production plan to its specific product. One of the first steps in that customization process is to review the process described in this section and to eliminate any steps that do not apply. For example, if the product will be built totally from existing services and devices, neither architecture definition nor component development is needed. Most of the work is required on the plan's schedule and bill of materials (described in Section 3.7); the schedule includes a specific timeline, while the bill of materials includes specific costs.

#### **3.6.1 Product Production**

The requirements for a specific product are analyzed and mapped to a set of core assets that are rated to produce the required system qualities. The set of assets is analyzed using the prediction tool to confirm that the resulting product will possess the required qualities. The product developers then assemble the product by writing sufficient "glue" code to interface the components. This "glue" is particularly necessary for vendor-supplied drivers.

### **3.7 Management Information**

#### **3.7.1 Bill of Materials**

The bill of materials for a specific product comprises several sections. The first section prices the CNS and its accompanying software. Subsequent sections price the hardware devices and supporting software, and the software services. The final three sections list the total hardware and software costs for a product, and the total product cost. This structure is illustrated in Table 3.

The cost of each interface device is divided into the cost of acquiring any interfacing hardware, which will usually be accompanied by a software driver, and the additional cost of software development to adapt or replace the software driver. The bill should also include the cost of the software for each service included in the product.

*Table 3: Bill of Materials for a Connect'Em Product*

Item Description	Unit Cost	Quantity	Quantity x Unit Cost
CNS	Central server cost Server software cost		
Interface device1	Hardware cost Software cost		
Interface device2	Hardware cost Software cost		
...	Hardware cost Software cost		
Service software1	Software cost		
Service software2	Software cost		
...	Software cost		
Total product hardware cost			
Total product software cost			
Total product cost			

The unit cost for hardware devices will directly reflect the actual expense of purchasing the device. The unit software cost for each software driver will be computed by identifying how many of the product line products will include the device. The expected sales volume for each product and the number of products will provide the total number of uses of each software component. The cost of the driver is the development, acquisition, or licensing cost. Thus the software unit cost is computed as shown in Equation 1. The total product cost is the sum of the total product hardware and software costs.

$$\text{Software Unit Cost} = \frac{\text{Cost of Driver}}{\text{number of products} \sum_{i=1} \text{projected unit sales per product}_i} \quad (1)$$

The bill of materials must be accurate, because it is the basis on which licensing fees owed to suppliers will be computed. It is also a planning tool for the product line managers, giving them an accurate record of outside obligations and a means of budgeting internal resources for developing core assets. Projected costs are updated as estimates of projected sales, costs of goods, or estimates of the resources needed to produce a useable driver change.

### 3.7.2 Production Resources

In addition to the core assets described in Section 3.3, the following resources are needed for producing a service package or a product that integrates several services:

- **personnel:** Service package teams will need personnel with development experience to modify or create the drivers, and to develop the service's logic and any service views/controls that are required for the various output devices. The team will also need personnel who have experience in testing with an emphasis on integration testing.
- **tools:** The PECT-based tool discussed in Section 3.4.3 is used to evaluate a specific configuration. In addition, a combinatorial testing tool is required to specify the most efficient means of testing sets of configurations [McGregor 01].
- **manufacturer's documentation:** The product developers will need access to the specifications of the hardware interface and the system that is to be controlled.

### 3.7.3 Schedule

Connect'Em determines the schedule for each product development by considering the number of atomic use cases<sup>5</sup> and the steps of the product development process that are required. Two configurations of the development process have been calibrated and can be used to accurately estimate the amount of effort required for a single atomic use case for each configuration. The schedule is computed using the type of product (which determines the process configuration) and the number of atomic use cases for the given product.

### 3.7.4 Product-Specific Details

A low-end electronics manufacturer has determined that there is a market for developing drivers for older appliances. The system-specific production plan for the Econo-HIS product would include a modification that allows the inclusion of these devices in that product.

### 3.7.5 Metrics

The product development team collects and retains data, including the following metrics about the product:

- number of atomic use cases
- percentage of product from core assets

---

<sup>5</sup> An atomic use case is one that has been factored out in a structured use case model [McGregor 98]. One organization may use several different types of these use cases. They are standard enough that each one can be completed using the same amount of development resources.

Connect'Em collects the following metrics about the product line:

- *schedule deviations from estimates*: Both overruns and underruns of the schedule will be identified and used to identify inaccuracies in the estimation process.
- *defect rates in core assets*: The defect-tracking system will be used to collect reports of defects in any core asset. Periodically the core asset builders will aggregate the information and compute defect intensities by asset.
- *modifications to core assets*: Each modification that is made to a core asset will be recorded.

---

## **4 Protect'Em: A Home Security Company**

Protect'Em is a small 25-person company that has been developing and installing home security systems for over 5 years. Its products include systems that manage a variety of devices, such as door and window sensors, glass breakage detectors, electronic door locks, and motion detectors. The systems also provide a range of responses to the detection of a security threat, such as sounding an alarm, turning on lights, and notifying the police.

Protect'Em is a regional company that competes with the bigger national companies by focusing on the needs of its customers and delivering customer-specific solutions rather than mass-market products. It competes on the basis of excellent customer service rather than price. Its products are sold directly to homeowners; the company offers basic and high-end products to meet a range of homeowners' budgets. It also installs the systems and provides maintenance services.

The company has produced a product line of home security products for the past five years and has considerable expertise in wiring houses for security systems and connecting and managing multiple security devices. Protect'Em now realizes that there is a business opportunity in the broader HIS market and that its core expertise permits entry into other domains beyond security.

Protect'Em has a small team of developers, all in the same location, which has enabled it to take a lightweight approach to its product line practices and respond quickly to customers' needs. However Protect'Em's willingness to provide customized solutions for its customers comes at a cost—the current architecture that supports Protect'Em's security products isn't very configurable, so each new customer system requires a unique architectural solution. The company would like to use the existing architecture as a baseline from which customer systems are derived, but the reality is that too much effort is expended on creating product-specific architectures. In addition, Protect'Em cannot afford to hire lots of new people to expand into areas beyond security, so it will need to plan the launch of its expanded product line carefully. The basic production strategy will be to incrementally roll out new features while retaining backward compatibility with existing products.

### **4.1 Production Plan Differences**

Protect'Em's business goal of providing excellent service to its customers means that customizability is the highest priority quality attribute (Section 2.2 of the production plan outline in the appendix) addressed in its production process. Protect'Em is confident that its production process for the security product line will scale up to the expanded HIS product

line (Section 4 of the plan). Because of Protect'Em's small size and budget, it cannot absorb the up-front cost of creating a comprehensive product line architecture for a complete line of HIS products. Therefore, the existing architecture for the security product line is the initial baseline for all products (Section 3.1 of the plan). Protect'Em has plans to address the full HIS domain eventually, but its limited resources dictate its present strategy. It established some basic rules for customizing the architecture and components (Section 4 of the plan), and the strategy (Section 2 of the plan) is to expand the current security architecture incrementally with each new paying customer.

Protect'Em tailors its current architecture and components for each new customer in a production process that bears little resemblance to the assemble-and-configure approach of Connect'Em. Its production plan must identify and coordinate all the changes to the baseline architecture (Section 4.2 of the production plan). It must also deal with a CM process (Section 5 of the plan) that is more complex than that of a proactive product line organization. Every new customer request generates new items to be placed under CM, and these variants might, in turn, put Protect'Em's existing products at risk.

Customization means that the rules for modifying the architecture and components must be made explicit in the production plan (if they are not already documented in the attached processes and incorporated by reference in the production plan). The product developer needs to know the rules for customization (Section 6 of the plan), and the process has to guarantee that those rules are followed.

The cost and schedule estimates in Protect'Em's production plan (Section 7 of the plan) are less predictable than those in Connect'Em's plan because of Protect'Em's willingness to customize. The metrics in the plan reflect the company's bias towards customer satisfaction: more effort is expended on collecting data to reduce product defects than on measuring product development costs and streamlining the production process. Metrics that should be included in Protect'Em's production plan include

- time taken to assemble a product once a specific product has been identified
- number of requests for changes to existing core assets to meet customization needs
- number of times an asset is used to build products. Protect'Em needs to "do more with less," so low levels of reuse mean greater effort elsewhere, most likely in the "glue" code.
- amount of "glue" code that needs to be written to integrate the pieces of a product
- granularity of reuse. Protect'Em needs to reuse more than just device drivers.

Protect'Em's biggest challenge is to deal with the tension between the discipline required by the product line approach and the demands of customizing products in ways that are outside the existing customizability of the product line. The desire to meet customers' needs and time-to-market requirements may cause product builders to make their own product-specific modifications to core assets—modifications that might lead to uncontrolled variability and a

CM headache. Protect'Em's strategy for the HIS product line depends heavily on feedback from product builders to core asset developers, since, in effect, Protect'Em's fielded products are the basis for future core assets. That dependency and the fact that core asset developers may not have time to redesign the assets to meet the product builders' needs might degrade the product line over time.

Since Protect'Em is broadening the scope of its product line incrementally by adapting the existing security architecture, the step in its product line development process that provides feedback from product builders to core asset developers is particularly important. This feedback reflects knowledge about the ease of customization, product-specific features that could/should be generalized and packaged as core assets, and the overall scope of the product line. The activity that actually obtains and records this knowledge could be specified as a feedback step in the production plan or as a proactive step in the development process for core assets.

---

## 5 Fleece'Em: A Home Automation Corporation

Fleece'Em is a large U.S. corporation that is the market leader for devices and associated software for a range of home automation applications. The company has developed several product lines that provide automated solutions for home security, safety, entertainment, heating and cooling, and smart appliances.

Fleece'Em's customers are wealthy homeowners who are willing to invest a significant amount of money in high-end home automation. They readily embrace new features and are early adopters of new technologies. There is a limited amount of variability in Fleece'Em's products, since most of its customers typically opt for complete, full-featured products rather than base systems to which extra features could be added incrementally.

"Technology Forecasting" is an important product line practice area for Fleece'Em. Its expertise in that area gives the company time to incorporate new devices and technological innovations into its core asset base smoothly. These, in turn, can be integrated smoothly into the production process, because they will have been created and tested properly in a product line context rather than as product-specific responses to new customer requirements. Fleece'Em has built its reputation on its ability to incorporate innovative new technologies into its products quickly.

Fleece'Em wants to "own" the domestic HIS market by complementing its leadership position in devices and services with a proprietary integration scheme that will link everything together. It has already established a domestic HIS product line and has two major business goals for it: (1) enter the global HIS market and (2) enter the fast-growing market for low-cost HIS solutions targeted to customers other than high-end HIS adopters. Fleece'Em thus faces two significant challenges: selling complete HISs in a worldwide market and moving beyond its traditional high-end customer base by scaling down its products to meet the needs of the low-cost market.

### 5.1 Production Plan Differences

The major difference between Fleece'Em's production plan (for its current, domestic HIS product line) and those of Connect'Em and Protect'Em is the high degree of automation that Fleece'Em applies to create products.<sup>6</sup> Fleece'Em's production process is simple: product developers identify the product to be built—for example, by selecting features (Section 4.1 of the production plan outline in the appendix)—and the automated support largely takes care of

---

<sup>6</sup> See the Product Gen variant of the Product Builder pattern [Clements 02].

the process of assembling the assets into a product (Section 4 of the plan). As mentioned above, there is not much variation to deal with since the high-end market demands feature-rich products. This simplifies Fleece'Em's integration testing (Section 4.5 of the plan) and CM (Section 5 of the plan). The exceptions are when a customer desires a combination of features not previously tested (no test report for this feature combination exists in Fleece'Em's testing database) or an entirely new feature. Additionally, Fleece'Em's stable asset base also means that its current production plan contains cost and schedule estimates (Section 7 of the plan) that are more reliable than Protect'Em's.

The future, however, is not so rosy. Fleece'Em's business goals are about to unleash major changes in its development process for HIS product lines. Fleece'Em wants to expand its HIS product line along two dimensions: (1) going global and (2) entering the low-cost market. The global aspect means that Fleece'Em will have to deal with cultural issues such as language and user interfaces. Also, it will have to address the coordination of business units and domain expertise distributed across different countries. There may be legal issues—*security* and *safety* may have very different legal interpretations and consequences in different countries. These, in turn, will affect the testing and integration of products, and the CM of country-specific product variations. System testing for particular countries may require that particular kinds of test hardware be used (e.g., keyboards and monitors for the Japanese market). If Fleece'Em is adept at managing its HIS product line, many of these issues can be resolved by the core asset developers, leaving the process of creating products unchanged. In reality, it is likely that going global will mean that problems will be solved in the short term by developers of country-specific products until Fleece'Em becomes better at handling country-specific variations in its product line.<sup>7</sup>

Fleece'Em's goal of repositioning itself as a player in the low-cost market is a difficult one for two reasons: (1) the current architecture is built to support the high-end market and (2) the partitioning of functionality and the associated qualities are not geared to the needs of the low-cost market. It will be difficult for Fleece'Em to extract and repackage smaller sets of features as low-cost products. Again, this is really a problem for the core asset developers that should, in theory, leave the automated production process unchanged.

Product identification will be harder, since there will be more features and greater variation in the ways in which they can be packaged into products. In addition, it will be harder to automate the generation and testing of products, and CM will be more complex. In fact, Fleece'Em's production plan will, in the short term, have to evolve into something much closer to the less automated schemes of Connect'Em and Protect'Em.

---

<sup>7</sup> It is worth noting that Fleece'Em's goal of forcing customers to use its proprietary home integration solutions (as opposed to, e.g., Connect'Em's open approach) does not affect its production plan. Product developers create products from assets; it's the development activities for core assets that are affected.

As a final comment on Fleece'Em, the second dimension of Fleece'Em's expansion strategy, entering the low-cost market, is markedly different from the strategies of Connect'Em and Protect'Em. These two organizations approach product building from the point of view of scaling up their existing production capability to expand into new markets. Fleece'Em has the opposite problem: scaling down. The one-size-fits-all strategy that worked so well for it in the high-end market is about to undergo a severe reality check.

---

## 6 Summary

This technical note demonstrates that how a product line organization builds products depends significantly on the organization's business goals, production strategy, and previous experience. The guidance on production planning provided by Chastek and McGregor is illustrated by discussing the productions plans of three example organizations [Chastek 02]. The fundamental problem of production planning is: What do product developers need to build a specific product and how do they do it? The examples highlight the different ways in which a production plan might address the problem and the influences that lead to specific courses of action.

Table 4 on the next page compares the significant production plan characteristics for the three hypothetical product line organizations. The numbers in the leftmost column correspond to the top-level sections of the production plan outline in the appendix.

The major discriminator of the three plans is the production strategy. That strategy is based on the business goals of the product line and has the greatest effect on production planning, because it coordinates the design and use of the assets that will be reused across the product line. The quality attributes of the production strategy (e.g., modularity and scalability) directly affect how products are created to meet the business goals.

The degree of automation applied to building products is also a significant driver of production planning. Fleece'Em's highly automated production process resembles the Product Gen variant of the Product Builder pattern, whereas the processes of the other two companies span all the practice areas of the full pattern [Clements 02].

*Table 4: Comparison of the Three Production Plans*

Plan Section	Connect'Em	Protect'Em	Fleece'Em
Introduction/ Context	has existing product lines and expertise in networking. Current customers are businesses.	small company with expertise in security systems. Current customers are homeowners.	large corporation with expertise in many HIS domains. Existing customers are high-end homeowners.
Strategy	Assemble and configure. Modularity and configurability are the most important qualities.	Incrementally roll out new features to reduce risk. Customizability is the most important quality.	Scale down existing high-end systems for low-cost market. Scalability is the most important quality.
Available Core Assets	OSGi-based architecture	existing security architecture	existing high-end HIS product line assets
Production Process	partially automated	partially automated	highly automated
CM	typical CM situation	complex CM because of extensive customization	simplified CM because of the product line's limited variability
Tailoring	simplified tailoring if the product is to be built wholly from existing services and devices	lots of product-specific customizations	simplified tailoring because of low variability of products and high degree of automation
Management Information	predictable schedule and cost estimates	Schedule and cost estimates are less predictable than Connect'Em's because of customization.	predictable schedule and cost estimates

The following outline of a production plan is based on Chastek and McGregor's work [Chastek 02].

- 1 Introduction
  - 1.1 Production Context
  - 1.2 Audience
  - 1.3 Qualifications
- 2 Strategic View of Product Development
  - 2.1 Assumptions
  - 2.2 Qualities
  - 2.3 Products Possible from Available Assets
  - 2.4 Production Strategy
- 3 Overview of Available Core Assets
  - 3.1 Basic Inputs and Dependencies
  - 3.2 Variations
- 4 Detailed Production Process
  - 4.1 Requirements Engineering
  - 4.2 Architecture Definition
  - 4.3 Architecture Evaluation
  - 4.4 Component Development
  - 4.5 Testing
  - 4.6 Software System Integration
- 5 Configuration Management
- 6 Tailoring the Production Plan
  - 6.1 Product Production
- 7 Management Information
  - 7.1 Bill of Materials

- 7.2 Production Resources
- 7.3 Schedule
- 7.4 Product-Specific Details
- 7.5 Metrics



---

## References

- [Bachmann 00]** Bachmann, Felix; Bass, Len; & Klein, Mark. *An Application of the Architecture-Based Design Method to the Electronic House* (CMU/SEI-2000-SR-009, ADA383836). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2000. <<http://www.sei.cmu.edu/publications/documents/00.reports/00sr009.html>>.
- [Chastek 01]** Chastek, Gary & Donohoe, Patrick. *Product Line Analysis: A Practical Introduction* (CMU/SEI-2001-TR-001, ADA396137). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2001. <<http://www.sei.cmu.edu/publications/documents/01.reports/01tr001.html>>.
- [Chastek 02]** Chastek, Gary & McGregor, John. *Guidelines for Developing a Product Line Production Plan* (CMU/SEI-2002-TR-006). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University. <<http://www.sei.cmu.edu/publications/documents/02.reports/02tr006.html>>.
- [Clements 02]** Clements, Paul & Northrop, Linda. *Software Product Lines: Practices and Patterns*. Boston, MA: Addison-Wesley, 2002.
- [Hissam 01]** Hissam, Scott; Moreno, Gabriel A.; Stafford, Judith; & Wallnau, Kurt. *Packaging Predictable Assembly with Prediction-Enabled Component Technology* (CMU/SEI-2001-TR-024, ADA399793). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2001. <<http://www.sei.cmu.edu/publications/documents/01.reports/01tr024.html>>.
- [HomePlug 02]** HomePlug Powerline Alliance. <<http://www.homeplug.org>> (valid as of September 2002).
- [Jacobson 99]** Jacobson, Ivar; Booch, Grady; & Rumbaugh, James. *The Unified Software Development Process*. Boston, MA: Addison-Wesley, 1999.

- [McGregor 98]** McGregor, John D. & Russ, Melissa. *A Qualitative Analysis of Two Requirements Capturing Techniques for Estimating the Size of Object-Oriented Software Projects* (TR-98-102). Clemson, SC: Clemson University, Department of Computer Science, 1998.
- [McGregor 01]** McGregor, John D. & Sykes, David A. *A Practical Guide to Testing Object-Oriented Software*. Boston, MA: Addison-Wesley, 2001.
- [OSGi 02]** Open Systems Gateway Initiative (OSGi), eds. *OSGi Service Platform (Release 2)*. Amsterdam, The Netherlands: IOS Press, 2002.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE September 2002		3. REPORT TYPE AND DATES COVERED Final	
4. TITLE AND SUBTITLE Product Line Production Planning for the Home Integration System Example			5. FUNDING NUMBERS F19628-00-C-0003	
6. AUTHOR(S) Gary Chastek, Patrick Donohoe, & John D. McGregor				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213			8. PERFORMING ORGANIZATION REPORT NUMBER CMU/SEI-2002-TN-029	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) HQ ESC/XPK 5 Eglin Street Hanscom AFB, MA 01731-2116			10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES				
12A DISTRIBUTION/AVAILABILITY STATEMENT Unclassified/Unlimited, DTIC, NTIS			12B DISTRIBUTION CODE	
13. ABSTRACT (MAXIMUM 200 WORDS) A production plan is a description of how a software product line organization builds products in a product line. This technical note examines the significant characteristics of the production plans of three hypothetical organizations that create product lines of home integration systems. Such systems enable homeowners to access and control equipment in their homes such as climate control and security systems. The plan for one of the organizations is presented in some detail, and the plans for the other two are described in terms of their differences from the first plan. The purpose of this note is to show how influences such as an organization's business goals, production strategy, and experience in product lines can lead to very different approaches to building products.				
14. SUBJECT TERMS software product line, production plan, product developer			15. NUMBER OF PAGES 40	
16. PRICE CODE				
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL	